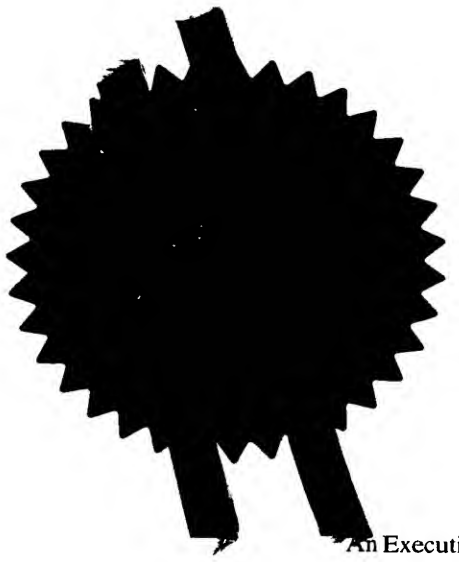
The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation and Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein together with the Statement of inventorship and of right to grant of a Patent (Form 7/77), which was subsequently filed.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.



Signed

Dated 19 June 2003

19JUL02 E734672-1 C59521
P01/7700 0.00-0216825.0

THE PATENT OFFICE
A
19 JUL 2002
NEWPORT

Patents Form 1/77

Patents Act 1977

Request for grant of a patent
Grant

1/77

The Patent Office
Concept House
Cardiff Road
Newport
Gwent NP10 8QQ

1.	Your reference	2034-P587-GB
2.	Patent application number	0216825.0 19 JUL 2002
3.	Full name, address and postcode of the or of each applicant *(underline all surnames)*	AUTODESK CANADA INC 10 Duke Street Montréal Quebec Canada H3C 2L7
	Patents ADP number *(if you know it)*	83780690001
	If the applicant is a corporate body, give the country/state of its incorporation	Quebec, Canada
4.	Title of the invention	Image Data Processing Apparatus
5.	Name of your agent	ATKINSON BURRINGTON
	"Address for service" in the United Kingdom to which all correspondence should be sent	25-29 President Buildings President Way Sheffield S4 7UR GB
	Telephone No:	0114 275 2400
	Patents ADP number	7807043001

6.	If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number	Country	Priority application number *(if you know it)*	Date of filing *(day/month/year)*
		N/A	N/A	N/A
7.	If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application		Number of earlier application	Date of filing (day/month/year)
			N/A	N/A
8.	Is a statement of inventorship and of right to grant of a patent required in support of this request?	Yes		

Patents Form 1/77

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	
Description	14
Claim(s)	04 CF
Abstract	01
Drawings	11 +11

10. If you are also filing any of the following, state how many against each item.

Priority documents	N/A
Translations of priority documents	N/A
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	None
Request for preliminary examination and search *(Patents Form 9/77)*	One
Request for substantive examination *(Patents Form 10/77)*	None
Any other documents *(Please specify)*	

11. I/We request the grant of a patent on the basis of this application.

Signature  Date **Thursday, 18 July 2002**

12. Name and daytime telephone number of person to contact in the United Kingdom

RALPH ATKINSON CPA
0114 275 2400

Patents Form 7/77

Patents Act 1977

7/77

Statement of inventorship and of right to grant of a patent

The Patent Office
Concept House
Cardiff Road
Newport
Gwent NP10 8QQ

1. Your reference — 2034-P587-GB

2. Patent application number — 02 16 825.0

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

AUTODESK CANADA INC
10 Duke Street
Montréal, Quebec
Canada H3C 2L7

4. Title of the invention — Image Data Processing Apparatus

5. State how the applicant*(s)* derived the right from the inventor*(s)* to be granted a patent

The applicant derived the right to the invention by virtue of contracts of employment

6. How many, if any, additional Patents Forms 7/77 are attached to this form? *(see note (c))* — None

7. I/We believe that the person*(s)* named over the page *(and on any extra copies of this form)* is/are the inventor*(s)* of the invention which the above patent application relates to.

Signature

Date Tuesday, 17 September 2002

8. Name and daytime telephone number of person to contact in the United Kingdom

RALPH ATKINSON CPA
0114 275 2400

Patents Form 7/77

Enter the full names, address and postcodes of the inventors in the boxes and underline the surnames

George Emmanuel BLIN
755 Moquin
Saint-Hubert
Quebec
Canada J3Y 6Y3

8466567001

Patents ADP number:

Le Huan TRAN
72 Parkdale Avenue
Pointe-Claire
Quebec
Canada H9R 3Y5

8209983002

Patents ADP number:

Patents ADP number:

Patents ADP number

Image Data Processing Apparatus

Background of the Invention

1. Field of the Invention

The present invention relates to image data processing apparatus and a method of processing image data.

2. Description of the Related Art

Image data processing systems for editing and manipulating moving image clips have been known for a number of years. Such systems contain secondary storage, typically in the form of a RAID (redundant array of independent discs), for the temporary storage of clips while the required editing process takes place.

A clip which is to be used in an editing process may comprise many megabytes of data. Therefore, if an attempt is made to load such a clip onto a RAID, and the available storage space on the RAID is insufficient, processing time and operator time is wasted. It is therefore known for systems to have a process for measuring the available storage space before the storage process is commenced. However, such measuring processes themselves require a substantial processing effort, and the corresponding processing time can impact on the workflow of the human operator.

Brief Summary of the Invention

According to a first aspect of the present invention there is provided an image data processing apparatus, comprising: receiving means for

receiving image frame data defining image frames of a predetermined definition; image frame storage means having a plurality of defined storage elements, such that each element is configured to store a defined quantity of said image frame data; memory means containing usage data indicating which of said storage elements is currently being used; and processing means configured to update said usage data in response to image frame data being stored within said image frame storage means, wherein said memory means further contains a data store, and said processing means is configured to: analyse said usage data to determine the number of said storage elements currently not being used; store data within said data store indicating the number of image frames of said predetermined definition which may be received by the non-used storage elements; and in response to receiving a request to store further image frames of said predefined definition, read data from said data store to determine whether said further image frames may be stored.

Brief Description of the Several Views of the Drawings

Figure 1 shows a system **100** for editing image data;

Figure 2 shows computer **101** of *Figure 1*;

Figure 3 shows disc storage array **105** of *Figure 1*;

Figure 4 shows an illustration of a striping process for storing image frames;

Figure 5 shows an illustration of three partitions **501**, **502** and **503** co-existing on the RAID **105**;

Figure 6 shows an example of a bitmap **601** representing a partition;

Figure 7 illustrates eight data caches **701** to **708**;

Figure 8 shows a flow chart illustrating the operation of the system **100**;

Figure 9 shows the step **803** of acting upon a request to load image frames onto the frame store;

Figure 10 shows the step **901** of determining available RAID space in further detail;

Figure 11 shows the step **806** of acting upon a request to delete image frames from the frame store **105**.

Written Description of the Best Mode for Carrying Out the Invention

A system **100** for editing image data is illustrated in *Figure 1*. The system is used by a human operator for editing, modifying, processing and adjusting video or film image data to form an output sequence that will eventually be stored onto digital tape.

The system comprises a computer **101,** such as an Octane manufactured by Silicon Graphics Inc., a monitor **102**, a graphics tablet **103** to allow the user to interact with a graphical user interface presented by the monitor and a keyboard **104** to facilitate alphanumeric input.

The system **100** further comprises a disc based frame storage system, referred to herein as a frame store **105**, and in preparation for image editing, images from one or more film or video clips are transferred to the frame store **105** from a digital tape player (not shown). The digital

tape player may be local to the system **100** or remote, with the transfer taking place via a network. The frame store **105,** comprises several high capacity hard disc drives, configured as a RAID (redundant array of independent discs), arranged to supply and store image data in parallel across several individual drives at once.

Using the frame store **105**, it is possible to play back and record high resolution film images, or video images, at any location in a clip without having to wait for a tape wind mechanism to reach the required frame, thereby facilitating non-linear editing. Furthermore the frame store facilitates real time play and record of image data.

The computer **101** includes a CD-ROM drive **106**, allowing program instructions to be loaded onto a hard disc within the Octane, from a CD-ROM **107**.

Figure 2

Computer **101** of *Figure 1* is detailed in *Figure 2*. The computer includes one or more programmable processing devices **201** that communicate with a system memory **202**, a local disc storage device **203**, a first interface **204** for communicating with the disc storage array **105** and a second interface **205** for communicating with the keyboard **104**, touch tablet **103** and stylus.

Processing device **201** operates in response to program instructions read from system memory **202**. On initiation, program instructions are loaded into the system memory **202** from the local disc **203**. Local disc **203**

receives program instructions via data storage media such as a CD ROM **107** receivable within a CD ROM reader **106**.

The system memory **202** contains kernel memory **206** which is reserved for the use of the operating system for computer **101**, and also data storage required by the present invention.

Figure 3

Disc storage array **105** is detailed in *Figure 3*. In this example, the array has a total of four magnetic disc drives **301** to **304**, each with a storage capacity of four hundred gigabytes. Image data is received from the processing system **101** and is supplied to the processing system **101** over a SCSI interface **305** or, alternatively, a fibre channel interface (not shown). Interface **305** communicates with a SCSI controller which in turn communicates with the individual drives over a channel **306**.

The individual frames stored on the frame storage system **105** form contiguous clips, usually derived from computer animation systems, video sources or cinematographic film sources. The frames are therefore arranged to be displayed at a particular display rate, such as thirty frames per second for NTSC, twenty-five frames per second for PAL or twenty-four frames per second for cinematographic film. Storage system **105** is therefore configured to allow these different types of frames to be transmitted at display rate or at multiples of display rate.

In addition to being displayed at different rates, and therefore requiring differing data transfer rates, the actual sizes of frames also varies

for different frame formats. Thus, for example a frame of NTSC video or PAL video requires approximately one megabyte. High definition television systems require an ever-greater degree of storage capability per frame and systems capable of processing images derived from cinematographic film may yet require a greater degree of storage per frame. The system therefore needs to be configured to allow frames to be transported at selected display rates and at selected frame definitions.

The frame storage system is optimised by dividing each image frame into a plurality of stripes and then writing each stripe to an individual disc storage device. In this way, data defining an image frame is distributed over three of the discs **301** to **303**. In addition, a further disc storage device **304** is required for parity data where similar bits within each stripe are XORed together to produce a parity stream that is written to the redundant disc. In this way, the loss of data from any one disc may be reconstituted by performing the XORing process for all the remaining data. Further details of such a system are given in the present Assignee's United States Patent No. 6,118,931.

Figure 4

A striping process for storing image frames is illustrated in *Figure 4*. An incoming frame **401** is divided into three stripes, identified as stripe zero, stripe one, and stripe two. A storage control process **402** performs an XOR operation to generate parity data. Thereafter, data is written in parallel to discs **301** to **304**. Thus in this example, disc **301** receives data from stripe

two, disc **302** receives data from stripe one, disc **303** receives data from stripe zero, disc **304** receives the parity data. The addressing of data from the stripes may identify substantially similar locations but with the application of an appropriate off-set. Thus, data is read from stripe one at the same locations as data being read from stripe zero but with an appropriate off-set as identified by arrow **403**.

Having established a system of using four discs to stripe image frames as shown in *Figure 4*, applications executed by processing system **101** may access the storage device but from the perspective of the application executed by processing system **101**, the four grouped drives operate as a single logical volume.

As mentioned above, the frame store **105** is used to process image frames of various predetermined definitions and display rates. In each case, a striping process similar to that described above may be used, resulting in each frame being stored as one or more stripes on each of the discs **301** to **304**. However, due to the varying quantity of data required to define images of varying definition, the lengths of the stored stripes depends upon the definition of the respective images. For example, both NTSC and PAL images may be stored using the above described process, but the NTSC images will require the storage of slightly shorter stripes than for the PAL images. Consequently, in order to store the data efficiently, image frames are stored in separate partitions on the RAID, with each partition configured to accept image data relating to image frames of a predetermined definition.

Higher definition images may also be stored using the striping

process described above. However, if the image definition is too large for image data to be written to and read from the frame store at the required rate, one or more additional RAIDs similar to frame store **105** may be used. In such a case, the RAIDs are used in conjunction, such that each frame is striped across the discs of each array.

Figure 5

An illustration of three partitions **501**, **502** and **503** co-existing on the FRAME STORE **105** is shown in *Figure 5*. Each of the four discs **301** to **304** is similarly partitioned and configured to store image frames having three different defined definitions A,B and C. Consequently, partition **501** has storage elements A0, A1, A2 etc. of a first size, chosen for the efficient storage of stripes from frames of definition A, partition **502** has storage elements B0, B1, B2 etc. configured to receive stripes from frames of definition B, and partition **503** has storage elements C0, C1, C2 etc. configured to receive stripes from frames of definition C. Thus, on a request to store image frames of a particular one of said definitions, the storage process **402** writes data into the respective partition.

It should be understood that each partition, such as partition **501** extends across all discs in the RAID **105** and a data storage element, such as element A10 in partition **501**, comprises of four storage blocks, one on each of the four discs, with each block being at the corresponding location on each disc.

In an alternative embodiment, in which the frame store comprises two

or more such RAIDs used in conjunction, a partition may extend across all discs in all of the RAIDS, or a group of discs. Thus, for example lower resolution image frames may be stored on one four disc RAID while higher definition frames are stored on the eight discs of two RAIDs. In this embodiment, each of the partitions contains a certain number of storage elements, but the elements from some partitions comprise blocks from four discs while storage elements of the high definition partitions may contain blocks from eight or more discs.

Figure 6

In order to manage the storage of image frames to the frame store **105**, the processor **201** utilises usage data which is stored within kernel memory **206**. The usage data takes the form of one or more bitmaps, which are arranged to represent the status of the storage elements within the partitions of the RAID. Each bit within the bitmap corresponds to just one of the storage elements, and is arranged to indicate whether or not a storage element is currently being used to store image data. Consequently, if a particular storage element is not currently being used, the corresponding bit is set to zero, and if it is being used it is set to one. In order to maintain the accuracy of the bitmap data, the storage control process **402** updates the bitmap as image data is written to and deleted from the frame store **105**.

An example of a bitmap **601** representing a partition is shown in *Figure 6*. The bitmap **601** may, for example, represent partition **501** of *Figure 5*. A small section **602** of the bitmap **601** is shown as enlarged

portion **603**. As illustrated by the enlarge portion **603**, the bitmap merely comprises ones and zeroes, indicating the current use, or non-use respectively, of the corresponding storage element within partition **501**.

Figure 7

A human operator of system **100** may need to process image data from many different clips, each comprising frames of one of several different predetermined image definitions. In order to edit, manipulate, etc. said clips, they are first stored onto the frame store **105**.

Thus, during an editing process the processor **201** receives a plurality of requests each requesting that that a clip comprising image data is to be stored on frame store **105**. If the storage control process merely attempted to store the clip there would be a possibility that the available storage space would be filled before all data defining the clip were stored, and thus the storage process would be unsuccessful.

Therefore, before commencing the storage of a clip, the processor **201** determines whether the frame store **105** contains sufficient free storage space to store said clip. This may be done by parsing the relevant bitmap to determine how many frames of the required image definition could be stored in the partition. If it is determined that sufficient storage space exists, then the clip is stored. If sufficient storage space does not exist then an appropriate message may be displayed to the user.

Due to the large storage capacity of the FRAME STORE **105**, and the corresponding large size of the bitmap, or bitmaps, the parsing process

may take as much as two or three seconds.

In order to improve workflow of the system **100**, the kernel memory **206** includes eight data caches **701** to **708** inclusive, as illustrated in *Figure 7*. Each of the data caches is configured to receive a frame definition, or size, and the corresponding number of storage elements which are currently available. Thus, for example, cache **701** may contain a frame definition, or size, of 1000, indicating NTSC frames and 10890, indicating that 10890 storage elements are currently not being used.

After parsing a bitmap as described above, the processor **202** stores the result to one of the caches. Consequently, if the storage request cannot be serviced, due to lack of storage space, the cached data becomes of use to further storage requests. That is, if a further request is received to store a clip of the same frame definition, then instead of parsing the bitmap, the necessary information can be read immediately from the cache, and a few seconds may be saved in processing time and human operator's time.

Thus, each of the caches **701** to **708** provide a data store which, when used, contain data indicating the number of image frames of a predetermined definition which may be received by non-used locations of the frame store.

Figure 8

A flow chart providing a simplified illustration of the operation of the system **100** is shown in *Figure 8*.

Following power on at step **801**, the system is initialised at step **802**.

During the initialisation, the operating system is booted up, hardware drivers are initialised, the editing application program is started, and bitmaps representing storage usage within the frame store **105** and the corresponding caches are initialised.

At step **803** a question is asked to determine whether a request has been received to load image frame data onto the frame store **105**. If this question is answered yes, then the request is acted upon at step **804**, before step **805** is entered. A second question is asked at step **805**, to determine whether a request has been received to delete image frames from the frame store. If this question is answered yes, then the request is acted upon at step **806** before step **807** is entered.

At step **807** image data is edited, manipulated etc. in response to input signals received from the keyboard **104**, or graphics tablet **105**. At step **808** it is determined whether or not a request has been received, indicating that the editing session is to be terminated. If this question is answered yes then the system may be shut down at step **809**, otherwise step **803** to **808** are repeated.

Of course, the order in which steps **803** to **807** are performed may be varied from the flow chart of *Figure 8*, which is merely intended to provide a simple example.

Figure 9

The step **803** of acting upon a request to load image frames onto the frame store is shown in more detail in *Figure 9*. At step **901**, the storage

control process determines the available free storage space on the frame store **105** for storing image frames of the required definition. At step **902** a question is asked to determine whether the number of frames which may be stored, as determined at step **901**, is greater than or equal to the number of frames requested to be stored. If it is determine at step **902** that the frame store has insufficient storage space, then an appropriate message is displayed on monitor **102** and step **804** is completed.

Alternatively, if step **902** finds that there is sufficient storage space, then, at step **903**, images frames are stored to the appropriate partition of the frame store **105** in accordance with the request. The relevant bitmap is updated at step **904** to reflect the changes made at step **903** to the storage space usage. At step **905** the corresponding cache **701**, **702**, **703**, **704**, **705**, **706**, **707** or **708**, is flushed, and the flushed cache is then updated at step **906**, to complete step **804**.

Step **906** of updating the cache may be performed by re-parsing the relevant bitmap. Alternatively, the new cached value may be calculated from the flushed value and the number of frames stored at step **903**.

Figure 10

The step **901** of determining available RAID space is shown in further detail in *Figure 10*. At step **1001** a question is asked to determine whether the available RAID space for image frames of the required definition has already been calculated and stored within one of the eight caches. If so, then the required information is read from the relevant cache

at step **1004**, and step **901** is completed. Otherwise, if the question asked at step **1001** is answered negatively, then the appropriate bitmap is parsed at step **1002** to determine how many storage elements are presently not being used, and hence how many frames of the requested image definition may be stored. The frame definition and number of frames which may be stored is then cached at step **1003** to complete step **901**.

Figure 11

The step **806** of acting upon a request to delete image frames from the frame store **105** is shown in more detail in *Figure 11*. At step **1101** frames are deleted from the relevant partition of the frame store **105**, in accordance with the request received at step **805**. At step **1104** the bitmap corresponding to said partition is updated, and then the corresponding cache is flushed and updated at steps **1105** and **1106** respectively. Thus steps **1104** to **1106** are, in essence, the same as steps **904** to **906**.

Claims:

1. Image data processing apparatus, comprising:

receiving means for receiving image frame data defining image frames of a predetermined definition;

image frame storage means having a plurality of defined storage elements, such that each element is configured to store a defined quantity of said image frame data;

memory means containing usage data indicating which of said storage elements is currently being used; and

processing means configured to update said usage data in response to image frame data being stored within said image frame storage means,

wherein said memory means further contains a data store, and said processing means is configured to:

analyse said usage data to determine the number of said storage elements currently not being used;

store data within said data store indicating the number of image frames of said predetermined definition which may be received by the non-used storage elements; and

in response to receiving a request to store further image frames of said predetermined definition, read data from said data store to determine whether said further image frames may be stored.

2. Image data processing apparatus according to claim **1**,

wherein said storage elements have a storage capacity configured to provide efficient storage of an image frame of said predetermined definition.

3. Image data processing apparatus according to claim **1** or claim **2**, wherein each said storage element has the storage capacity to store only one of said image frames of said predetermined definition.

4. Image data processing apparatus according to any of claims **1** to **3**, wherein said image frame storage means comprises a hard disc.

5. Image data processing apparatus according to any of claims **1** to **4**, wherein said image frame storage means comprises a RAID (redundant array of independent discs).

6. Image data processing apparatus according to any of claims **1** to **5**, wherein said usage data comprises a data element corresponding to each storage element.

7. Image data processing apparatus according to any of claims **1** to **6**, wherein said usage data comprises a bitmap.

8. Image data processing apparatus according to claim **7**, wherein said analysis of said usage data comprises parsing said bitmap.

9. Image data processing apparatus according to any of claims **1** to **8**, wherein said data store comprises a cache within kernel memory.

10. A method of processing image data, comprising the steps of:

receiving image frame data defining image frames of a predetermined definition;

storing said image frame data within image frame storage apparatus having a plurality of defined storage elements, such that each element is configured to store a defined quantity of said image frame data;

storing usage data indicating which of said storage elements is currently being used;

in response to image frame data being stored within said image frame storage means, updating said usage data;

analysing said usage data to determine the number of said storage elements currently not being used;

storing data within a data store indicating the number of image frames of said predetermined definition which may be received by the non-used storage elements; and

in response to receiving a request to store further image frames of said predefined definition, reading data from said data store to determine whether said further image frames may be stored.

11. A computer-readable medium having computer-readable instructions executable by a computer such that, when executing said

instructions, a computer will perform the steps of:

receiving image frame data defining image frames of a predetermined definition;

storing said image frame data within image frame storage apparatus having a plurality of defined storage elements, such that each element is configured to store a defined quantity of said image frame data;

storing usage data indicating which of said storage elements is currently being used;

in response to image frame data being stored within said image frame storage means, updating said usage data;

analysing said usage data to determine the number of said storage elements currently not being used;

storing data within a data store indicating the number of image frames of said predetermined definition which may be received by the non-used storage elements; and

in response to receiving a request to store further image frames of said predefined definition, reading data from said data store to determine whether said further image frames may be stored.

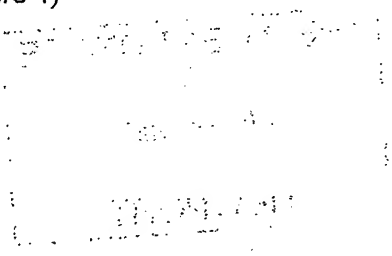
Abstract of the Disclosure

Image Data Processing Apparatus

Image data processing apparatus (**100**), comprising receiving means for receiving image frame data defining image frames of a predetermined definition; image frame storage means (**105**) having defined storage elements, such that each element is configured to store a defined quantity of the image frame data; memory means (**206**) containing usage data indicating which of the storage elements is currently being used; and processing means (**201**) configured to update the usage data in response to image frame data being stored within the image frame storage means.

The processing means is configured to analyse the usage data to determine the number of the storage elements currently not being used. Data is then stored, within a data store within memory means (**206**), indicating the number of image frames of the predetermined definition which may be received by the non-used storage elements. In response to receiving a request to store further image frames of the predetermined definition, data is read from the data store to determine whether the further image frames may be stored.

(Figure 1)



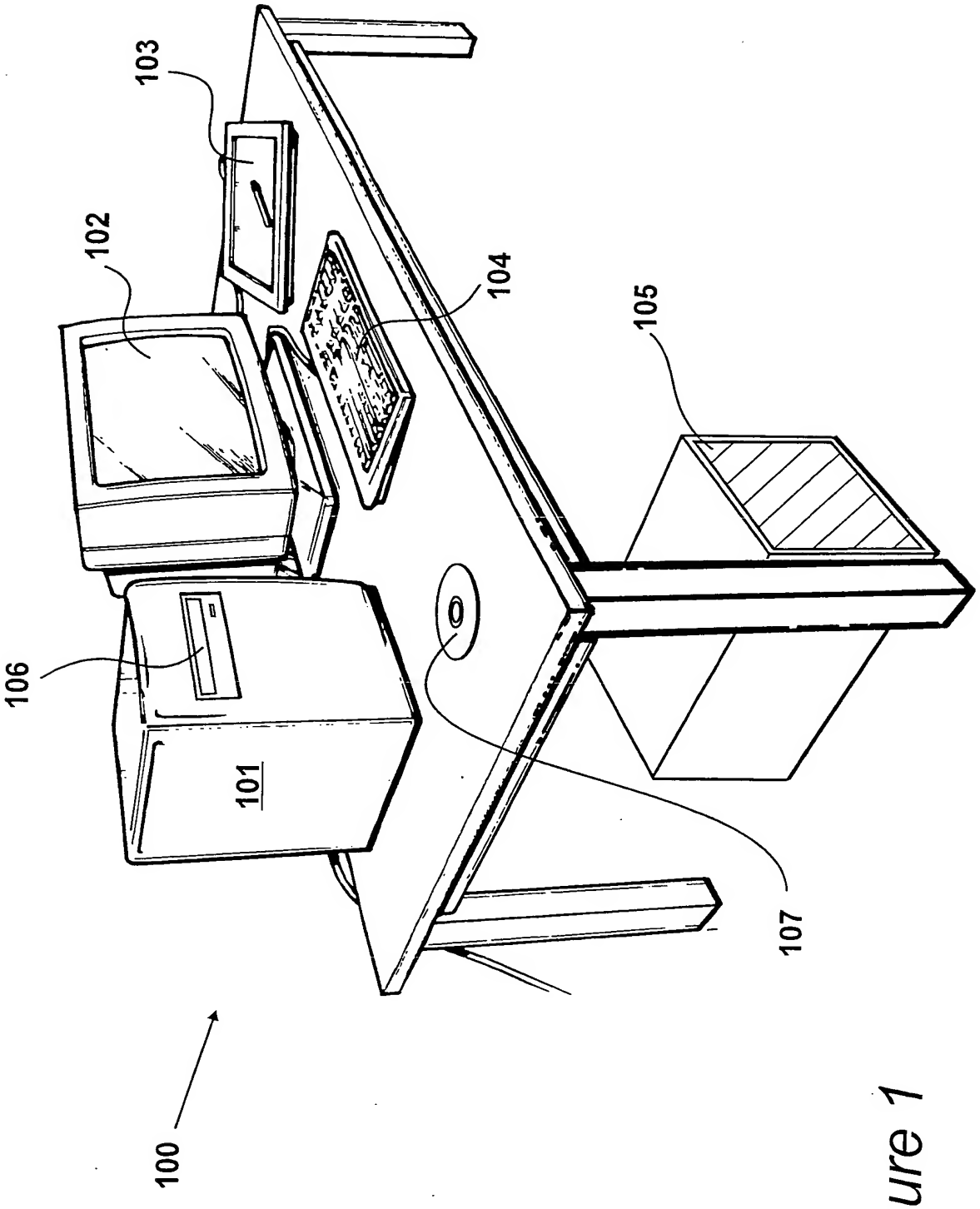


Figure 1

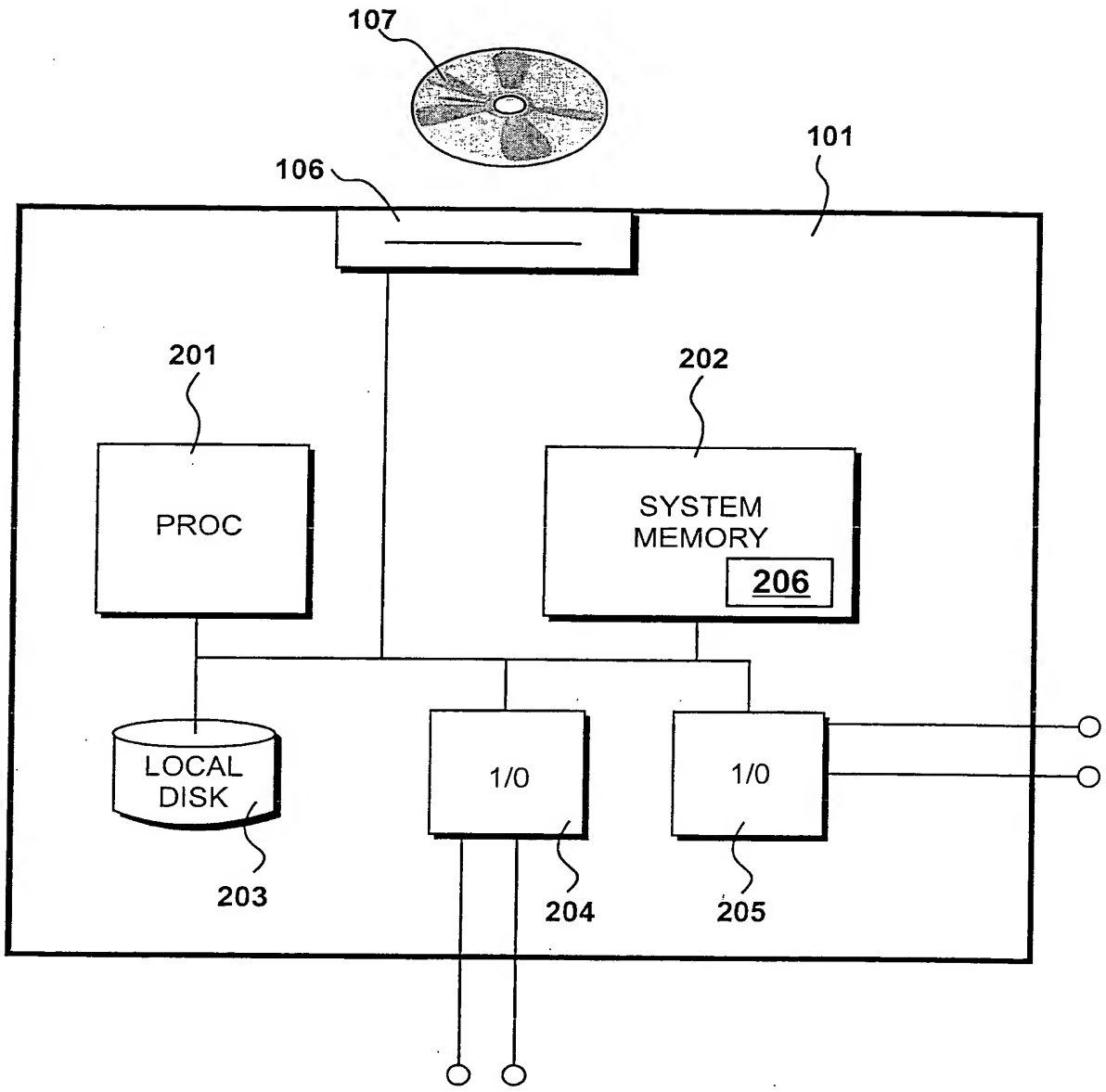

107
101
106
201
202
PROC
SYSTEM
MEMORY
206
LOCAL
DISK
1/0
1/0
203
204
205

Figure 2

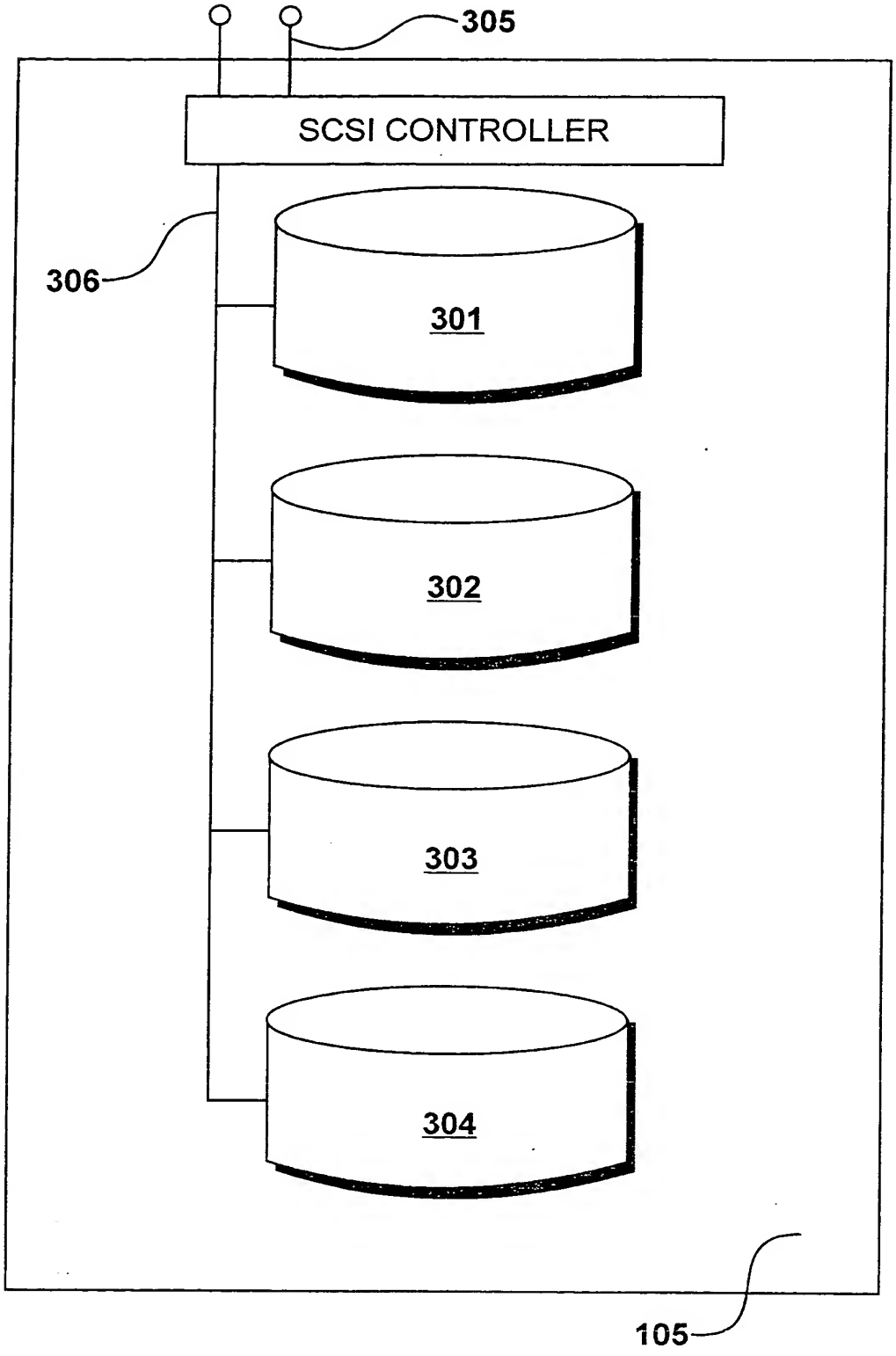


Figure 3

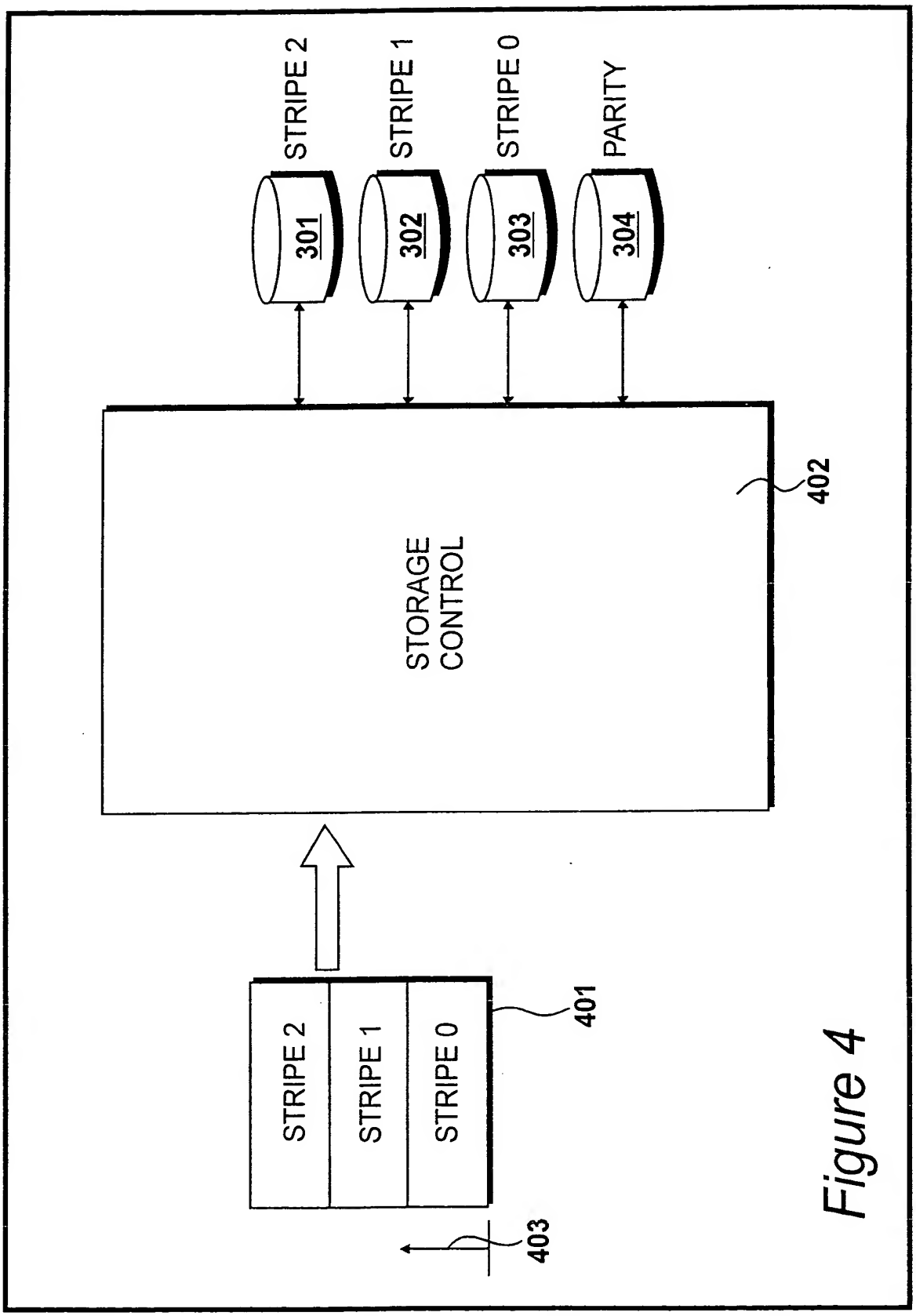

STRIPE 2
STRIPE 1
STRIPE 0
PARITY
301
302
303
304
STORAGE CONTROL
402
STRIPE 2
STRIPE 1
STRIPE 0
401
403

Figure 4

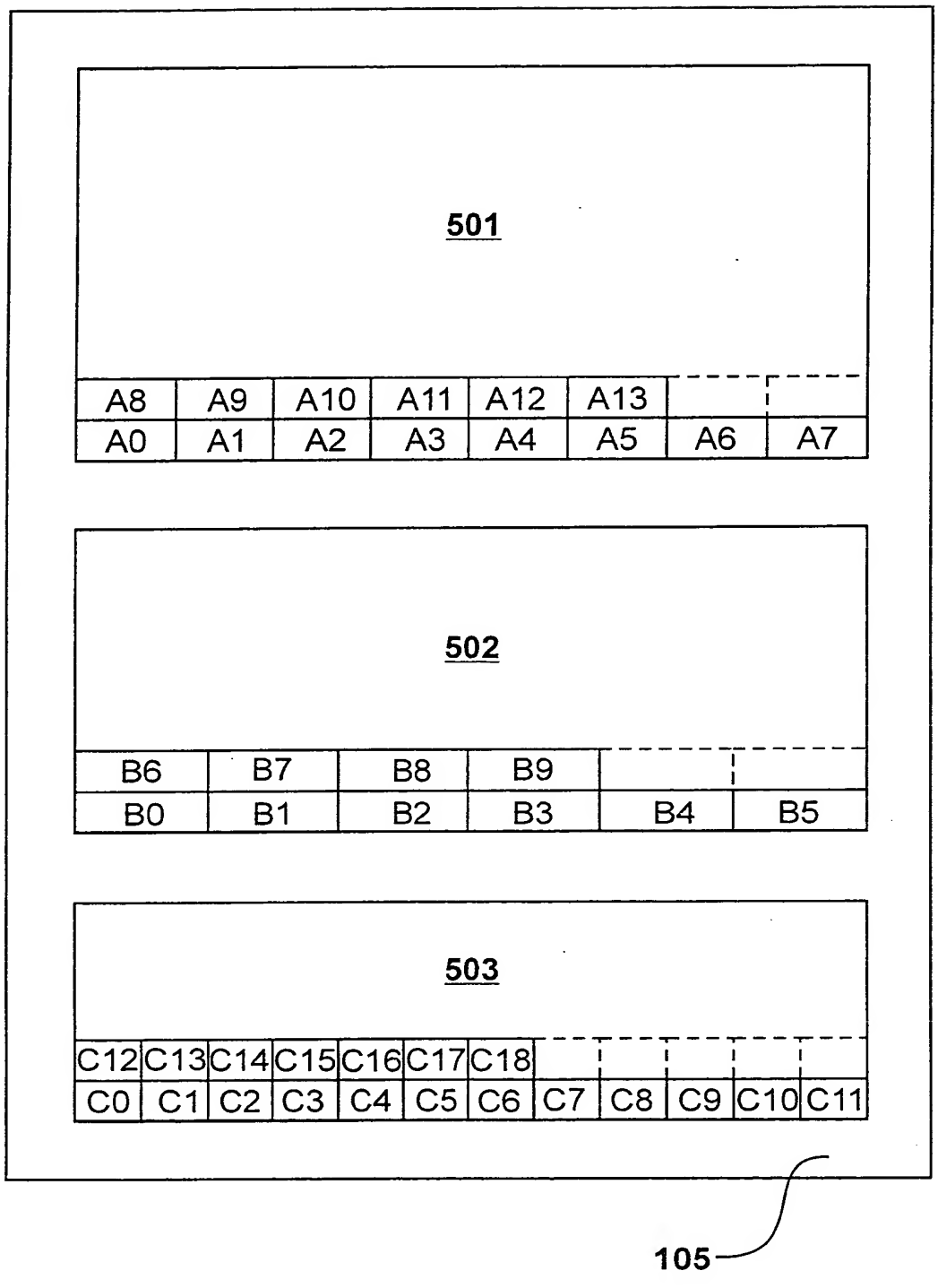

501
A8
A9
A10
A11
A12
A13
A0
A1
A2
A3
A4
A5
A6
A7
502
B6
B7
B8
B9
B0
B1
B2
B3
B4
B5
503
C12
C13
C14
C15
C16
C17
C18
C0
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
105

Figure 5

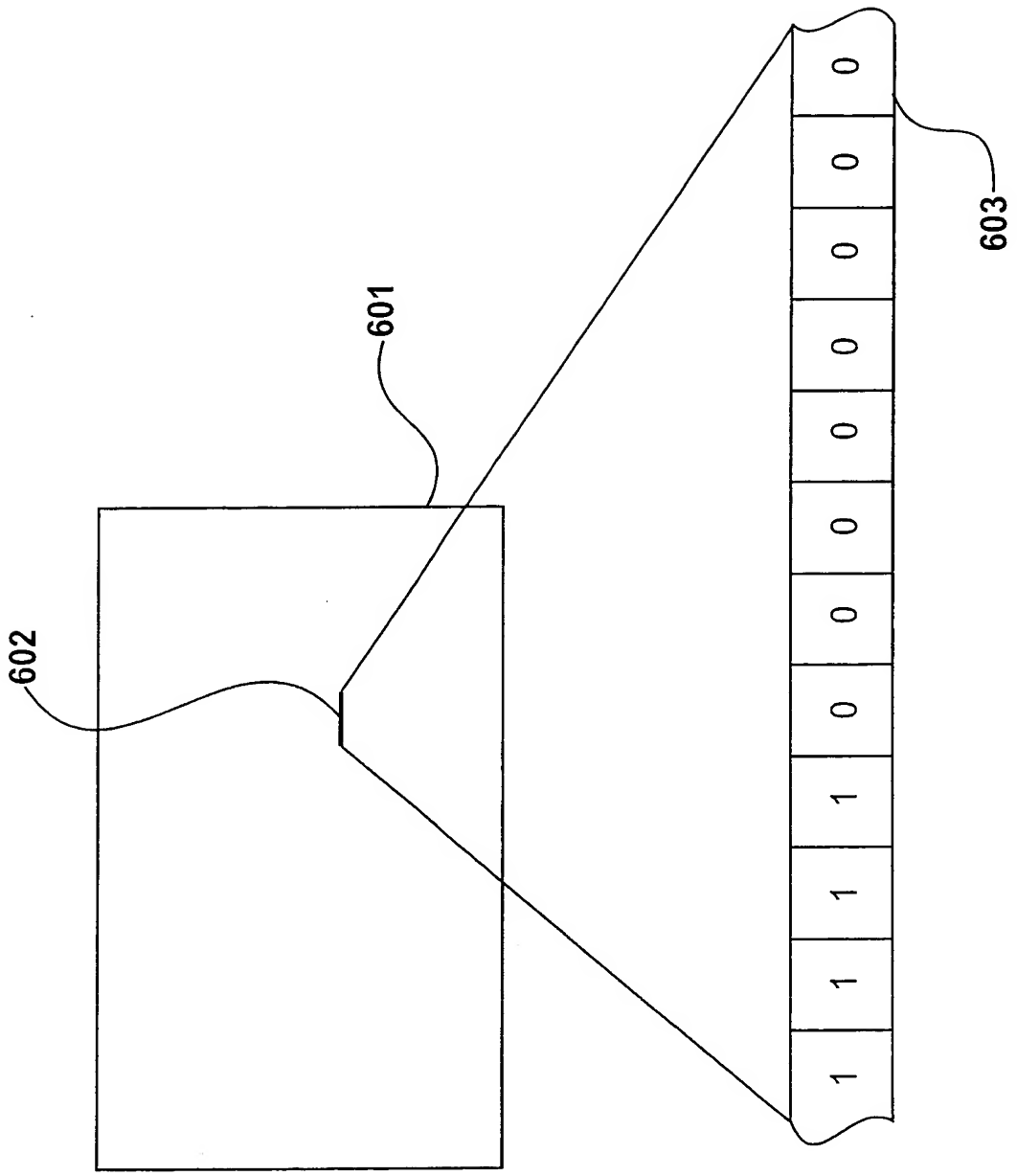


Figure 6

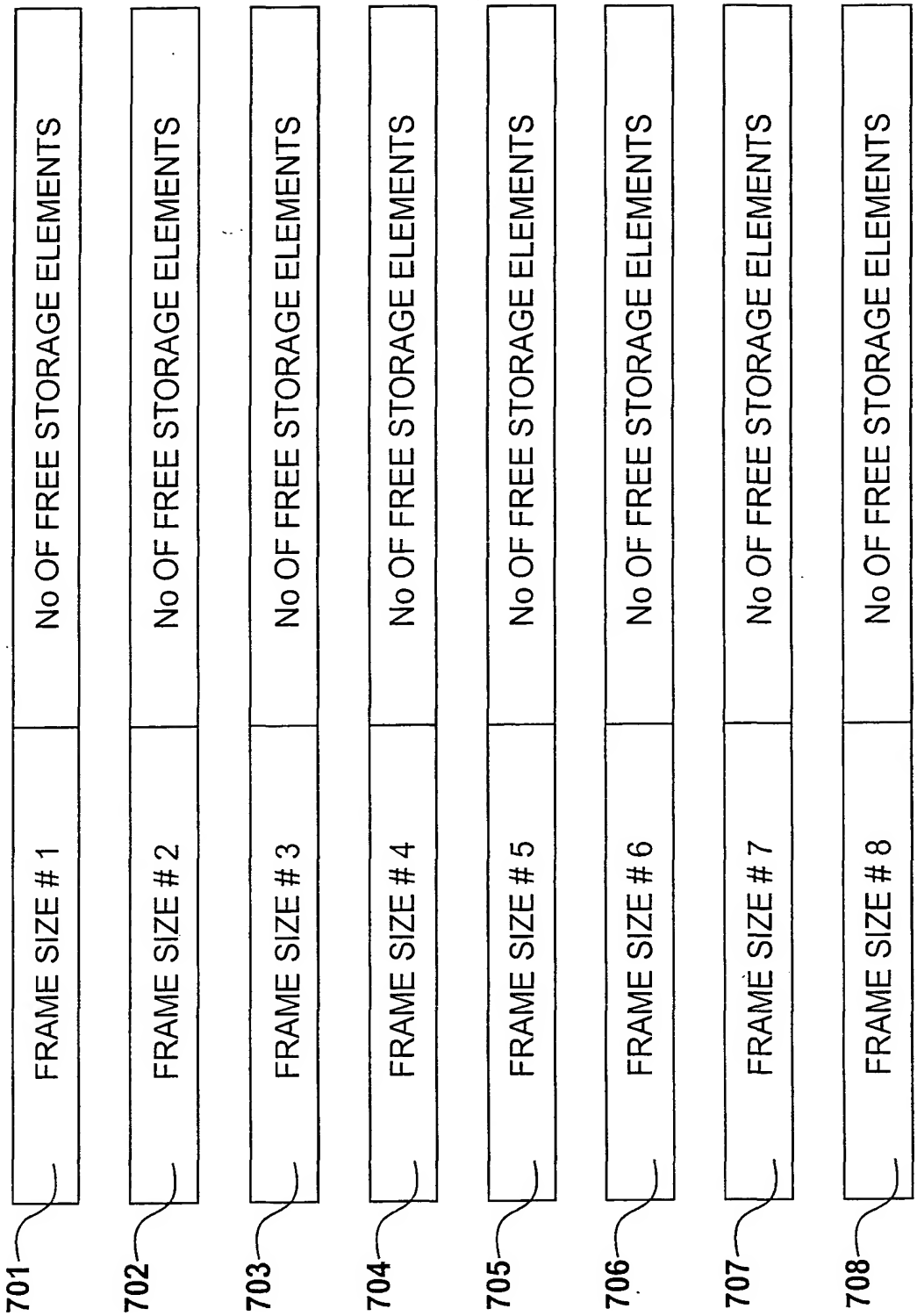

FRAME SIZE # 1
No OF FREE STORAGE ELEMENTS
FRAME SIZE # 2
No OF FREE STORAGE ELEMENTS
FRAME SIZE # 3
No OF FREE STORAGE ELEMENTS
FRAME SIZE # 4
No OF FREE STORAGE ELEMENTS
FRAME SIZE # 5
No OF FREE STORAGE ELEMENTS
FRAME SIZE # 6
No OF FREE STORAGE ELEMENTS
FRAME SIZE # 7
No OF FREE STORAGE ELEMENTS
FRAME SIZE # 8
No OF FREE STORAGE ELEMENTS
701
702
703
704
705
706
707
708

Figure 7

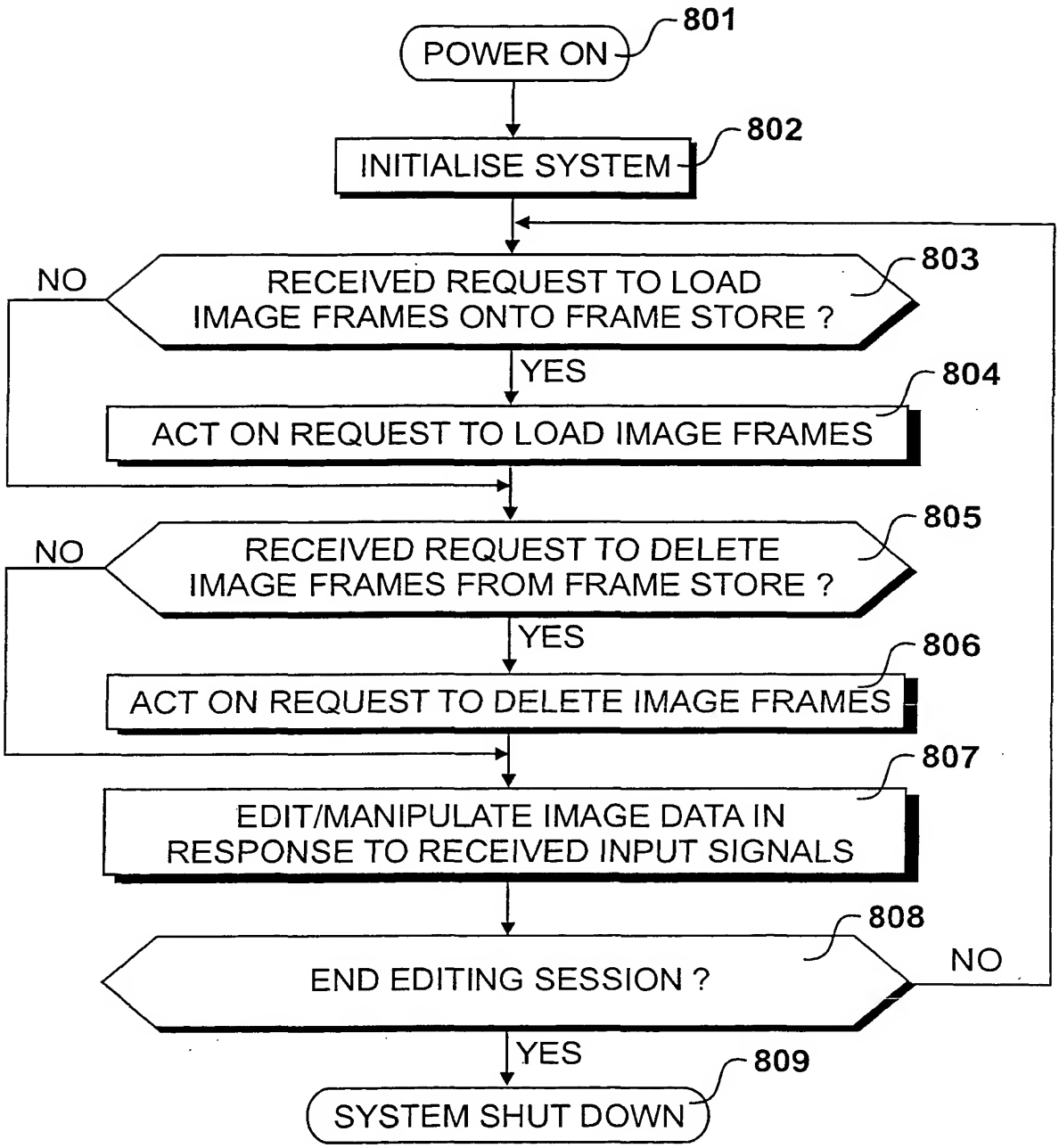

801
POWER ON
802
INITIALISE SYSTEM
803
RECEIVED REQUEST TO LOAD IMAGE FRAMES ONTO FRAME STORE ?
NO
YES
804
ACT ON REQUEST TO LOAD IMAGE FRAMES
805
RECEIVED REQUEST TO DELETE IMAGE FRAMES FROM FRAME STORE ?
NO
YES
806
ACT ON REQUEST TO DELETE IMAGE FRAMES
807
EDIT/MANIPULATE IMAGE DATA IN RESPONSE TO RECEIVED INPUT SIGNALS
808
END EDITING SESSION ?
NO
YES
809
SYSTEM SHUT DOWN

Figure 8

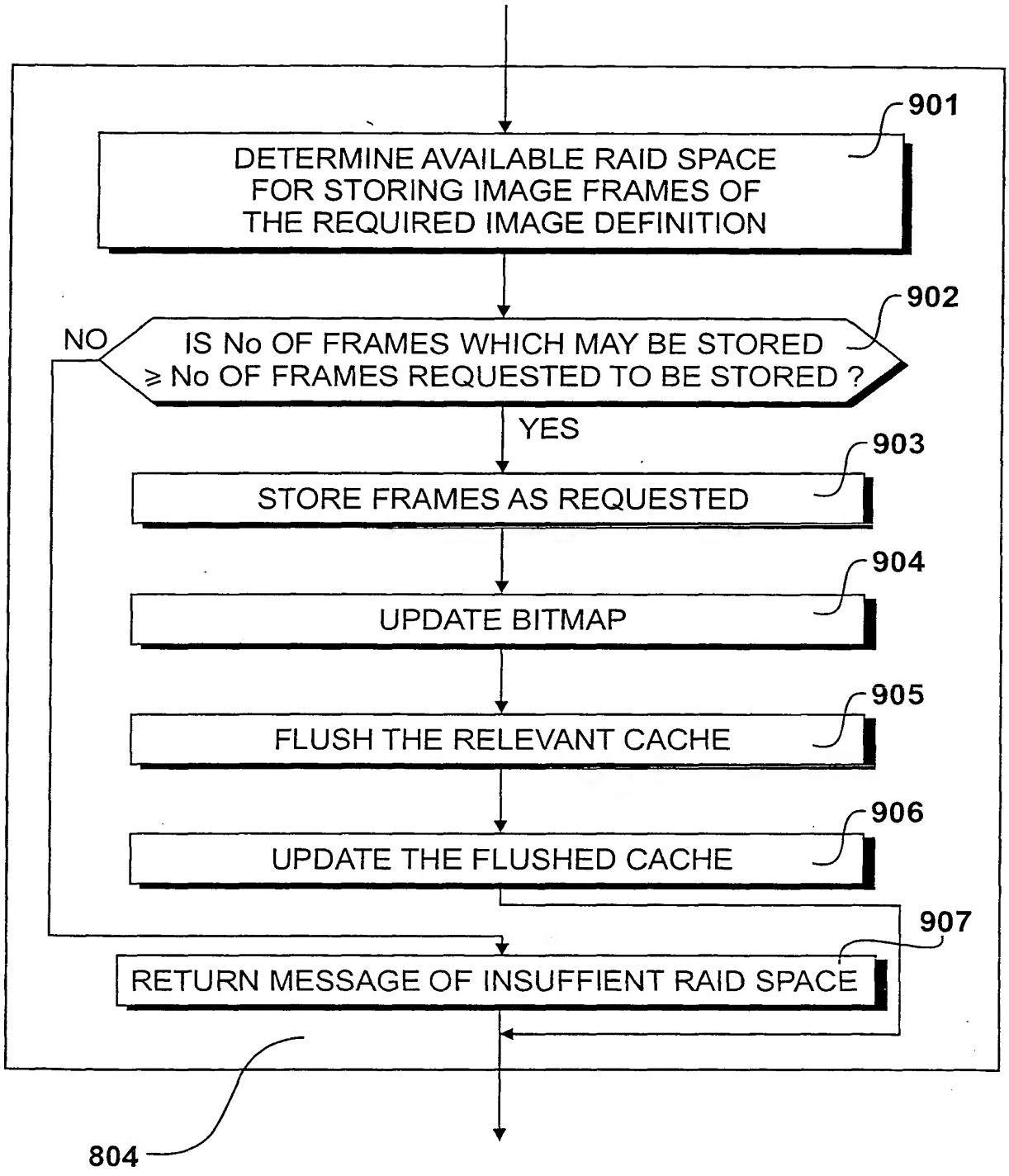

901
DETERMINE AVAILABLE RAID SPACE FOR STORING IMAGE FRAMES OF THE REQUIRED IMAGE DEFINITION
902
NO
IS No OF FRAMES WHICH MAY BE STORED ≥ No OF FRAMES REQUESTED TO BE STORED ?
YES
903
STORE FRAMES AS REQUESTED
904
UPDATE BITMAP
905
FLUSH THE RELEVANT CACHE
906
UPDATE THE FLUSHED CACHE
907
RETURN MESSAGE OF INSUFFIENT RAID SPACE
804

Figure 9

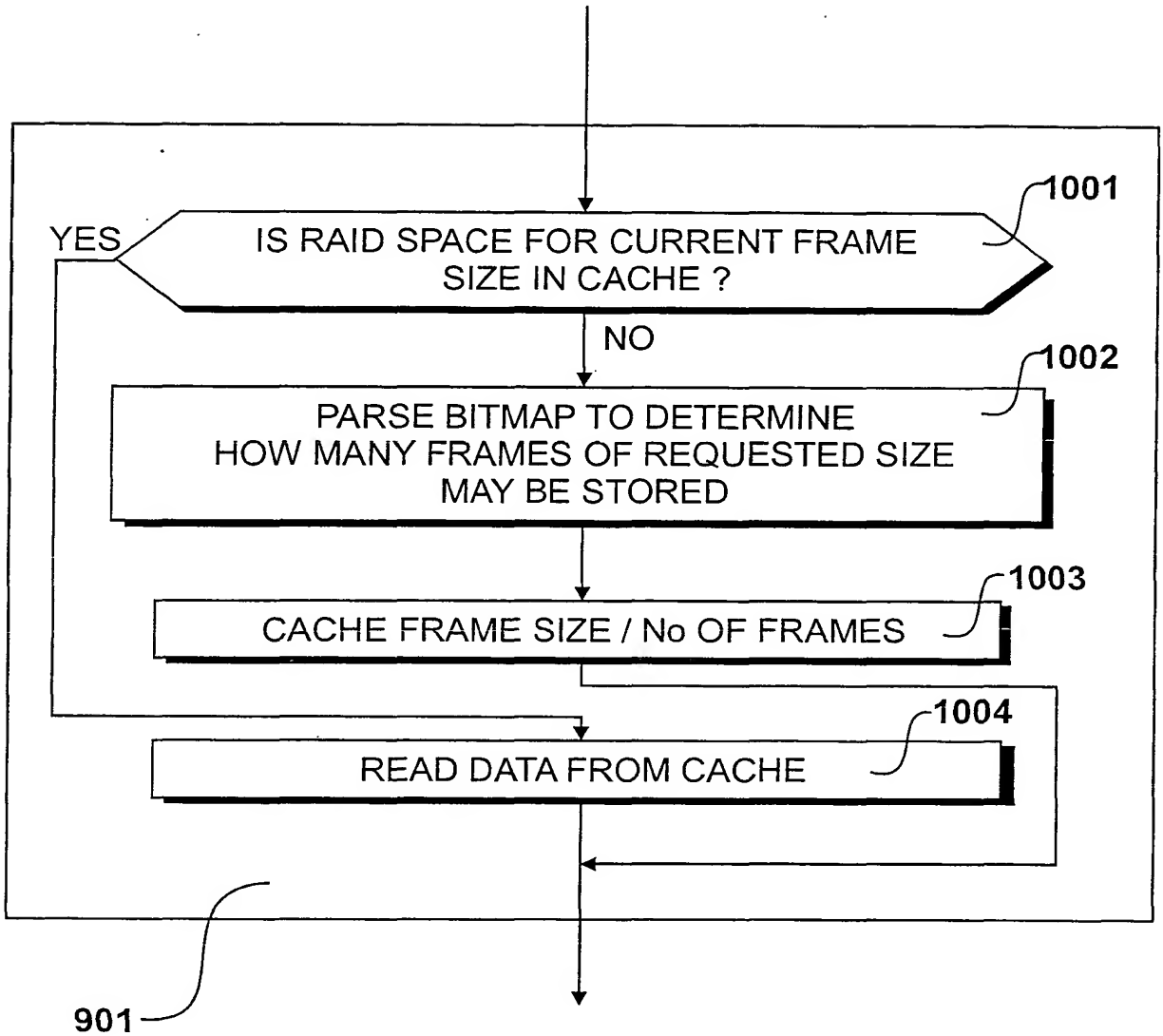

1001
YES
IS RAID SPACE FOR CURRENT FRAME SIZE IN CACHE ?
NO
1002
PARSE BITMAP TO DETERMINE HOW MANY FRAMES OF REQUESTED SIZE MAY BE STORED
1003
CACHE FRAME SIZE / No OF FRAMES
1004
READ DATA FROM CACHE
901

Figure 10

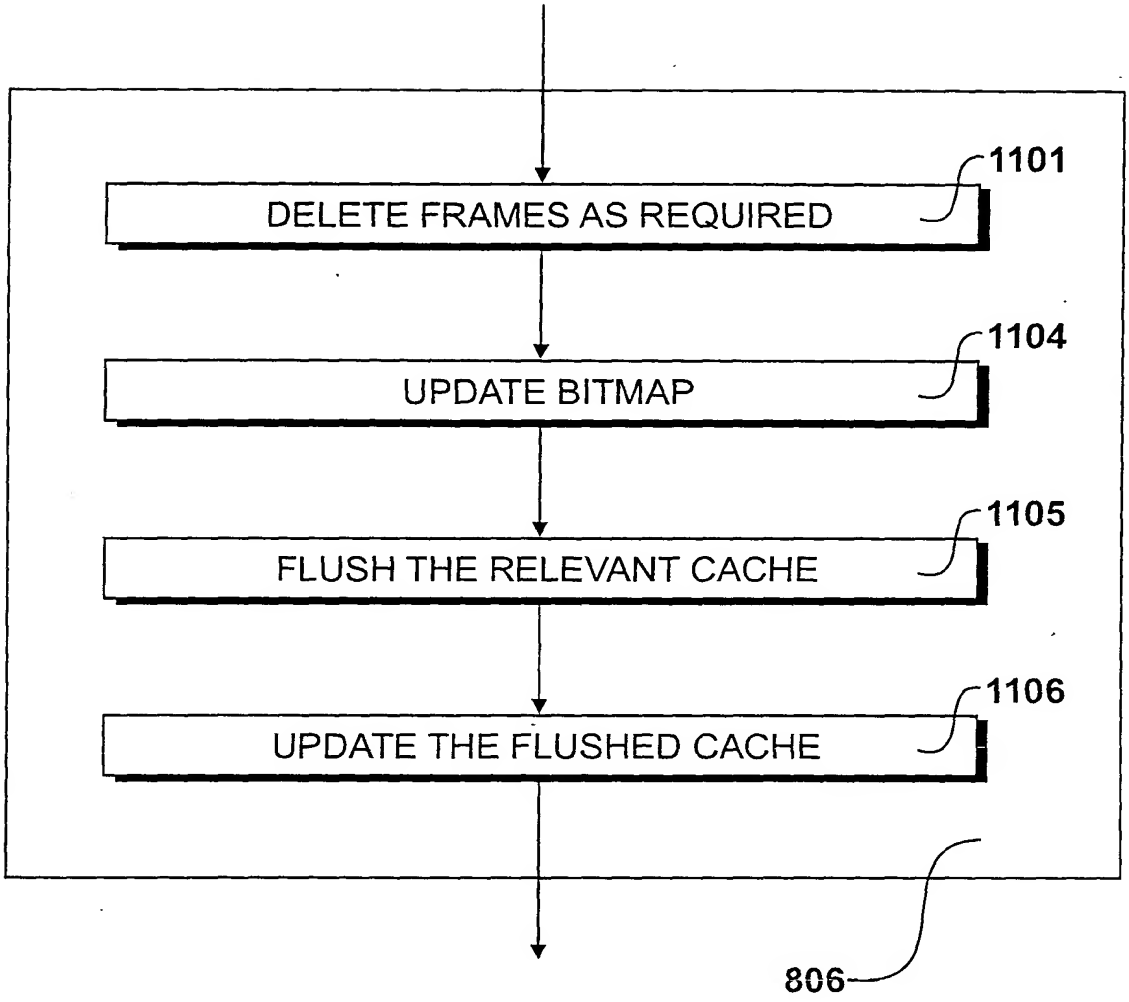

DELETE FRAMES AS REQUIRED
1101
UPDATE BITMAP
1104
FLUSH THE RELEVANT CACHE
1105
UPDATE THE FLUSHED CACHE
1106
806

Figure 11